# Kamera Tool

## Inhaltsverzeichnis


- 1 Vorbereitung
- 2 Kamera Tool im Spiel
- 3 Bilder Aufnehmen
- 4 Videos Aufnehmen
- 5 Mit FFmpeg Einzelbilder zu einem Video zusammenfügen
- 6 Alles noch ein mal in Videoform

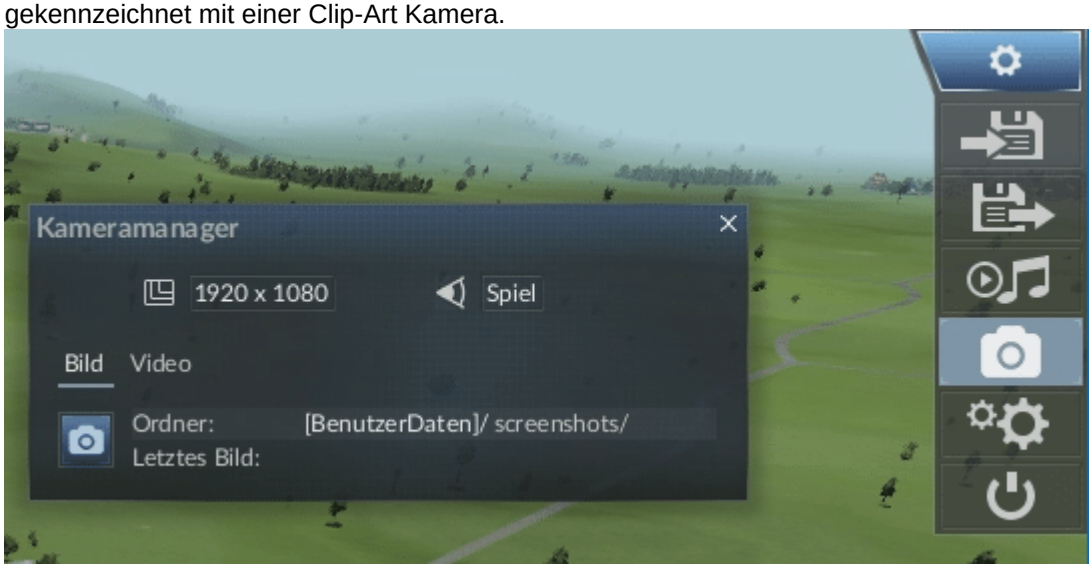

## 1 Vorbereitung

Um das Kamera Tool nutzen zu können, braucht ihr mindestens Build 12659 und der Debug Modus muss eingeschaltet sein.
Geht dazu im Hauptmenü unter Einstellungen zum Reiter Erweitert und schaltet den Debug Modus ein.



## 2 Kamera Tool im Spiel

Im Spiel findet ihr das Kamera Tool unter den Einstellungen am oberen rechten Bildschirmrand gekennzeichnet mit einer Clip-Art Kamera.




Die Steuerung im Kamera Tool unterscheidet sich von der normalen Steuerung im Spiel:

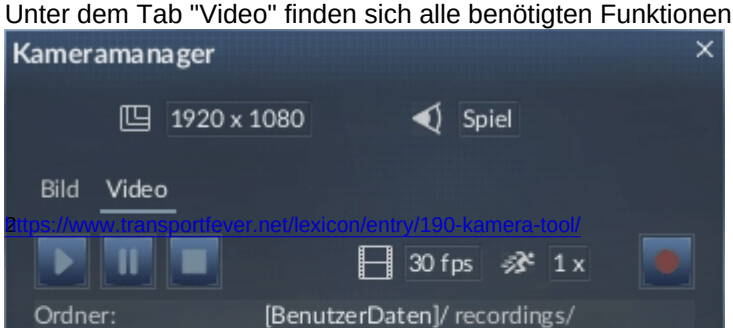
- Bei gedrückter linker Maustaste bewegt sich die Kamera vertikal
- Bei gedrückter rechter Maustaste bewegt sich die Kamera horizontal
- Bei gedrückter mittlerer Maustaste kann man die Sicht rotieren
- Mit dem Mausrad bewegt sich die Kamera nach vorne / hinten

## 3 Bilder Aufnehmen

Die Vorschau zeigt genau den Ausschnitt, der auch später im Bild verewigt wird.
Allerdings werden die Bilder mit deutlich höherer Qualität (alle Objekte im LOD0) und ohne die GUI aufgenommen.
Für Einzelbilder können Auflösung und Brennweite (Zoom) gewählt werden.
Mit einem Klick auf das Kamera-Symbol im Kameramanager wird das Foto als .png abgelegt.
Alle Bilder werden im Ordner `[BenutzerDaten]/screenshots/` abgelegt, wobei der Pfad zum [BenutzerOrdner] angezeigt wird, wenn die Maus über dem Pfad gehalten wird.

## 4 Videos Aufnehmen

Etwas komplexer als die Aufnahme von einem einzelnen Bild ist die Aufnahme von Bildserien, die zu einem Video zusammengefügt werden können.
Unter dem Tab "Video" finden sich alle benötigten Funktionen dafür:




2https://www.transportfever.net/lexicon/entry/190-kamera-tool/

Wie beim Einzelbild bewegt man zuerst die Kamera in die Position, die als erstes im Video zu sehen sein soll.
Durch einem Klick auf [+] oder mit [A] wird ein so genannter Keyframe (dt. Schlüsselbild) erstellt.
Mit mindestens zwei Keyframes kann ein Video erstellt werden.
Jeder Keyframe definiert die Position und Ausrichtung der Kamera, sowie die Brennweite und auch Bewegungsgeschwindigkeit der Kamera durch den definierten Punkt.
Jeder dieser Werte wird für alle Frames (Einzelbilder) zwischen den definierten Keyframes mittels Interpolation berechnet.

Zur Erstellung eines einfachen Videos werden zwei Keyframes definiert, mit einer leicht verschobenen Position und Ausrichtung zueinander.
Der Pfad der Kamera wird im Spiel als rote Linie dargestellt, mit der Ausrichtung der Kamera als schwarzer Linie bei jedem Keyframe.





Mit dem ? (Play) Symbol kann das Video in einer Vorschau gezeigt werden. Mit dieser Vorschau wird die Funktion der Keyframes und der Pfad der Kamera am einfachsten deutlich.
Wenn die Vorschau zufriedenstellen ist, können die definierten Keyframes mit einem Klick auf das Disketten-Symbol () für spätere Verwendung gespeichert werden.

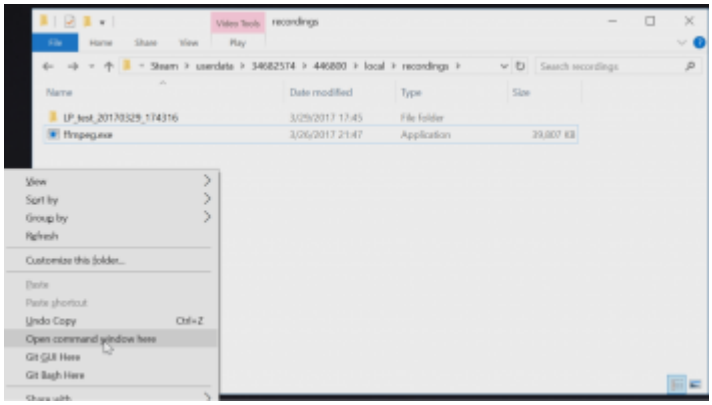
Nachdem die Auflösung und Framerate gewählt ist, kann die eigentliche Aufnahme mit einem Klick auf den roten Record-Knopf (?) gestartet werden.
Die einzelnen Frames werden aufgezeichnet und mit fortlaufender Nummer im Dateinamen als Einzelbilder im tga Format abgespeichert.

Alle Bilder von einem Video liegen in einem Unterordner, mit dem Namen der gespeicherten Keyframes und dem aktuellen Datum und Zeitstempel.

# 5 Mit FFmpeg Einzelbilder zu einem Video zusammenfügen

Um aus den Einzelbildern ein Video zu machen, kann ein beliebiges Videoprogramm benutzt werden.
FFmpeg ist ein Kommandozeilen-Programm, welches mit so ziemlich allen Video- & Audioformaten als Ein- und Ausgabe zurecht kommt und in vielen größeren Projekten "unter der Haube" verwendet wird.
Auf der FFmpeg Webseite kann man das kostenlose OpenSource Programm für Windows, Linux und Mac herunterladen - also alle Betriebssysteme, auf denen Transport Fever läuft. Für die meisten Linux-Distributionen wird ffmpeg in den offiziellen Paketquellen zur Verfügung gestellt und kann zum Beispiel mit `sudo apt install ffmpeg` direkt installiert werden. Die kompilierte Windows-Version ist hier zu finden. Nachdem das zip-Archiv heruntergeladen ist, wird die Datei "ffmpeg.exe" in den "recordings"-Ordner verschoben, in dem sich die Unterordner mit den Einzelbildern befinden. Unter Linux reicht die Installation, wodurch der Befehl `ffmpeg` automatisch zum "Path" hinzugefügt wird und aus jedem Ordner heraus benutzt werden kann.

Im recordings-Order wird nun ein Terminal / eine Kommandozeile geöffnet. Unter Windows hierzu "Shift" gedrückt halten und Rechtsklick in den Ordner im Windows Explorer -> Öffne Kommandozeile
Bei neueren Windows 10 Versionen wurde die Kommandozeile im Kontextmenü durch die Windows Power Shell ersetzt. Für uns bietet die Power Shell die selbe Funktionalität und wir können daher beides benutzen.




Mit dem folgenden Befehl kann eine Serie von Einzelbildern (mit fortlaufender Nummer) in ein mp4 Video umgewandelt werden:

Code

```
ffmpeg -framerate 30 -i "name_datum\name_img-%d.tga" -c:v libx264 -crf 20 -preset medium -pi
```

In der Windows Power Shell muss `ffmpeg` eventuell durch `.\ffmpeg` oder `.\ffmpeg.exe` ersetzt werden.

Die Einzelteile dieses Befehls im Detail:

- `ffmpeg`
  - Der Name des Programms. Falls der Befehl nicht gefunden wird, kann mit `.\ffmpeg` der aktuelle Ordner explizit angegeben werden oder mit `/pfad/zu/ffmpeg` der Pfad angegeben werden. Unter Windows kann noch die Dateiendung .exe angehangen werden `.\ffmpeg.exe`
- `-framerate 30`
  - Die FPS, mit der die eingelesenen Einzelbilder im Video abgespielt werden. Im Normallfall sollte dies der selbe Wert wie die eingestellten FPS bei Aufnahme der Frames in Transport Fever sein
- `-i "name_datum\name_img-%d.tga"`
  - Die Eingabedateien. Hier wird der Pfad zum ersten Bild angegeben (Ordner- und Dateinamen können mit "Tab" vervollständigt werden). Dann wird die "1" aus dem Dateinamen mit `%d` ersetzt.
  - Der Pfad kann zum Beispiel heißen: `test_20170329_174316\test_img-%d.tga`
- `-c:v libx264`
  - Der benutzte Videocodec: h264, hier können natürlich auch andere Codecs angegeben werden.
- `-crf 20`
  - Die Qualität des h264 codierten Videos (18 ist sehr gut, 23 ist normal, 28 ist eher schlecht)
- `-preset medium`
  - Komprimierungsfaktor im Verhältnis zur Kodiergeschwindigkeit
- `-pix_fmt yuv420p`
  - Pixelformat der Einzelbilder. Ohne diesen Parameter wird meist das Format YUV444 benutzt, womit ältere Player Probleme haben können. Für den normalen Gebrauch und größte Kompatibilität sollte YUV420 benutzt werden.

- `video.mp4`
  - Name der Ausgabedatei

Weiter Informationen zu den Parametern für ffmpeg finden sich im ffmpeg Wiki: https://trac.ffmpeg.org/

Die Einstellungen für den h264 codec sind hier zusammengetragen: https://trac.ffmpeg.org/wiki/Encode/H.264

**Tip für Nvidia Nutzer:** Viele Grafikkarten von Nvidia unterstützen das dekodieren und encodieren von h264 (und h265) Videos direkt in der GPU. Dies ist um ein vielfaches schneller, als ein Video allein mit der CPU umzuwandeln.
Der Nvidia Encoder `h264_nvenc` kann mittels ffmpeg angesprochen werden, und hat eben genau diesen

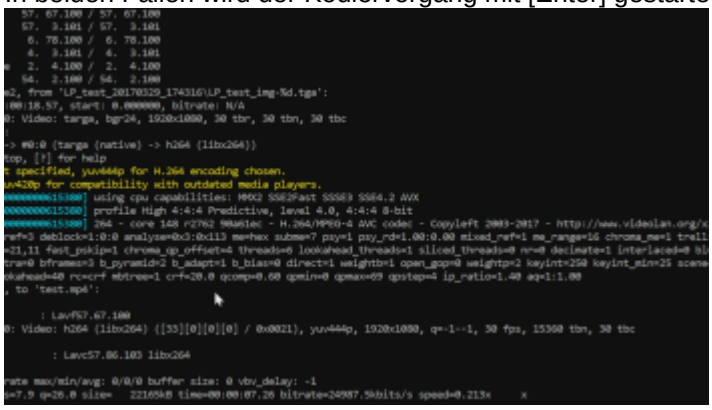
Vorteil. Bekannt ist die Kodierung von h264 Videos in der GPU einigen Nutzern sicher schon durch GeForce Shadowplay / Share oder der passenden Option im OBS.
Der Nvidia-basierte H264-Encoder benutzt leicht andere Parameter und sieht dann wie folgt aus:

Code

```
ffmpeg -framerate 30 -i "name_datum\name_img-%d.tga" -c:v h264_nvenc -qp 19 -preset medium
```

Weitere Informationen zum Encoder findet man natürlich im Netz, und mittels `ffmpeg -help encoder=h264_nvenc`

In beiden Fällen wird der Kodiervorgang mit [Enter] gestartet.



## 6 Alles noch ein mal in Videoform

Externer Inhalt youtu.be
Inhalte von externen Seiten werden ohne Ihre Zustimmung nicht automatisch geladen und angezeigt.
Alle externen Inhalte anzeigen
Durch die Aktivierung der externen Inhalte erklären Sie sich damit einverstanden, dass personenbezogene Daten an Drittplattformen übermittelt werden. Mehr Informationen dazu haben wir in unserer Datenschutzerklärung zur Verfügung gestellt.